

**SOFTWARE TESTING  
AND QUALITY ASSURANCE  
LAB MANUAL**

**TYBsc Computer Science  
(V Semester)  
For Academic Year (2024-25)**



## JNAN VIKAS MANDAL'S

PADMASHREE DR. R.T.DOSHI DEGREE COLLEGE OF  
INFORMATION TECHNOLOGY  
MOHANLAL RAICHAND MEHTA COLLEGE OF COMMERCE  
DIWALIMAA DEGREE COLLEGE OF SCIENCE  
AMRATLAL RAICHAND MEHTA COLLEGE OF ARTS  
JVM'S DEGREE COLLEGE OF INFORMATION TECHNOLOGY  
AIROLI, NAVI MUMBAI – 400708  
NAAC Reaccredited Grade 'A+' (CGPA- 3.31, 3<sup>rd</sup> Cycle)

## CERTIFICATE

This is to certify that the Mr./Miss. \_\_\_\_\_  
of T.Y.B.Sc.(CS) Semester-V has completed the practical work in the subject of  
**Software Testing And Quality Assurance** during the Academic year 2024-25  
under the guidance of **Dr. Sanjivani Nalkar** being the partial requirement for the  
fulfillment of the curriculum of Degree of Bachelor of Science in Computer  
Science, University of Mumbai.

**Place: Airoli**

**Date: 04<sup>th</sup> Oct 2024**

-----  
Sign of Subject In Charge

-----  
Sign of External Examiner

-----  
Sign of Incharge / H.O.D

# INDEX

Sr.No.	Name Of Practicals	Date	Signature
1	Selenium IDE Demo Installation	03/08/2024	
2	Conduct a test suite for any two web sites.	03/08/2024	
3	Install Selenium server and demonstrate it using a script in Java/PHP.	10/08/2024	
4	Write and test a program to login a specific web pages(Using JUnit)	23/08/2024	
5	Write and test a program to update 10 student records into table into Excel file (using TestNG )	14/09/2024	
6	Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).	21/09/2024	
7	Write and test a program to provide total number of objects present / available on the page.	28/09/2024	
8	Write and test a program to get the number of items in a list / combo box.	28/09/2024	

## **List of Practical's:**

1. Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats.
2. Conduct a test suite for any two web sites.
3. Install Selenium server (Selenium RC) and demonstrate it using a script in Java/PHP.
4. Write and test a program to login a specific web page.
5. Write and test a program to update 10 student records into table into Excel file
6. Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).
7. Write and test a program to provide total number of objects present / available on the page.
8. Write and test a program to get the number of items in a list / combo box.
9. Write and test a program to count the number of check boxes on the page checked and unchecked count.
10. Load Testing using JMeter, Android Application testing using Appium Tools, Bugzilla Bug tracking tools.

Practical :1

**Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats.**

**Solution:**

### **Introduction to Selenium**

#### **A. History of Selenium**

- In 2004 invented by Jason R. Huggins and team.
- Original name is JavaScript Functional Tester [JSFT]
- Open source browser based integration test framework built originally by Thought works.
- 100% JavaScript and HTML
- Web testing tool
- That supports testing Web 2.0 applications
- Supports for Cross-Browser Testing(ON Multiple Browsers). And multiple Operating Systems
- Cross browser – IE 6/7, Firefox .8+, Opera, Safari 2.0+

#### **B. What is Selenium?**

- Acceptance Testing tool for web-apps
- Tests run directly in browser
- Selenium can be deployed on Windows, Linux, and

Macintosh. Implemented entirely using browser technologies –

- JavaScript
- DHTML
- Frames

Selenium

Components-

- Selenium IDE
- Selenium Core
- Selenium RC

### C. Selenium IDE

- Selenium-IDE (Integrated Development Environment) is the tool you use to develop your Selenium test cases.
- It is Firefox plug-in
- Firefox extension which allows record/play testing paradigm
- Automates commands, but asserts must be entered by

hand Creates the simplest possible Locator Based on Selenese.

### D. Overview of Selenium IDE:

- Test Case Pane
- Toolbar
- Menu Bar
- Log/Reference/UI-Element/Rollup Pane
- Test Case Pane:
- Your script is displayed in the test case

pane. It has two tabs.-

One for displaying the command (source)and their parameters in a readable “table” format.

Command	<input type="text" value="selectWindow"/>	▼
Target	<input type="text" value="name=null"/>	<input type="button" value="Find"/>
Value	<input type="text"/>	

1. **Toolbar:** The toolbar contains buttons for controlling the execution of your test cases, including a step feature for Menu Bar
2. **File Menu:** The File menu allows you to create, open and save test case and test suite files.
3. **Edit Menu:** The Edit menu allows copy, paste, delete, undo and select all operations for editing the commands in your test case.
4. **Options Menu:** The Options menu allows the changing of settings. You can set the timeout value for certain commands, add user- defined user

extensions to the base set of Selenium commands, and specify the format (language) used when saving your test cases.

## **E. Introducing Selenium Commands-**

The command set is often called selenese. Selenium commands come in three “flavors”:

- Actions,
- Accessors
- Assertions.

**Actions:** user actions on application / Command the browser to do something. Actions are commands that generally manipulate the state of the application.

- Click link- click / Click and wait
- Selecting items

**Accessors:** Accessors examine the state of the application and store the results in variables, e.g. "storeTitle".

**Assertions:** For validating the application we are using Assertions

- For verifying the web pages
- For verifying the text
- For verifying alerts
- Assertions can be used in 3 modes:
- assert
- verify
- waitFor

Example: "assertText","verifyText" and "waitForText".

### **NOTE:**

- When an "assert" fails, the test is aborted.
- When a "verify" fails, the test will continue execution

- "waitFor" commands wait for some condition to become true
- Commonly Used Selenium Commands

These are probably the most commonly used commands for building test.

- **open** - opens a page using a URL.
- **click/clickAndWait** - performs a click operation, and optionally waits for a new page to load.
- **verifyTitle/assertTitle** - verifies an expected page title.
- **verifyTextPresent**- verifies expected text is somewhere on the page.
- **verifyElementPresent** -verifies an expected UI element, as defined by its HTML tag, is present on the page.
- **verifyText** - verifies expected text and it's corresponding HTML tag are present on the page.
- **verifyTable**-verifies a table's expected contents
- **waitForPageToLoad** -pauses execution until an expected new page loads. Called automatically when clickAndWait is used.
- **waitForElementPresent** -pauses execution until an expected UI element, as defined by its HTML tag, is present on the page.

## F. Recording and Run settings

- When Selenium-IDE is first opened, the record button is ON by default.
- During recording, Selenium-IDE will automatically insert commands into your test case based on your actions.
- Remember Base URL MODE - Using Base URL to Run Test Cases in Different Domains

**Record Absolute recording mode** – Run Test Cases in Particular Domain.

## Running Test Cases

- Run a Test Case- Click the Run button to run the currently displayed test case.



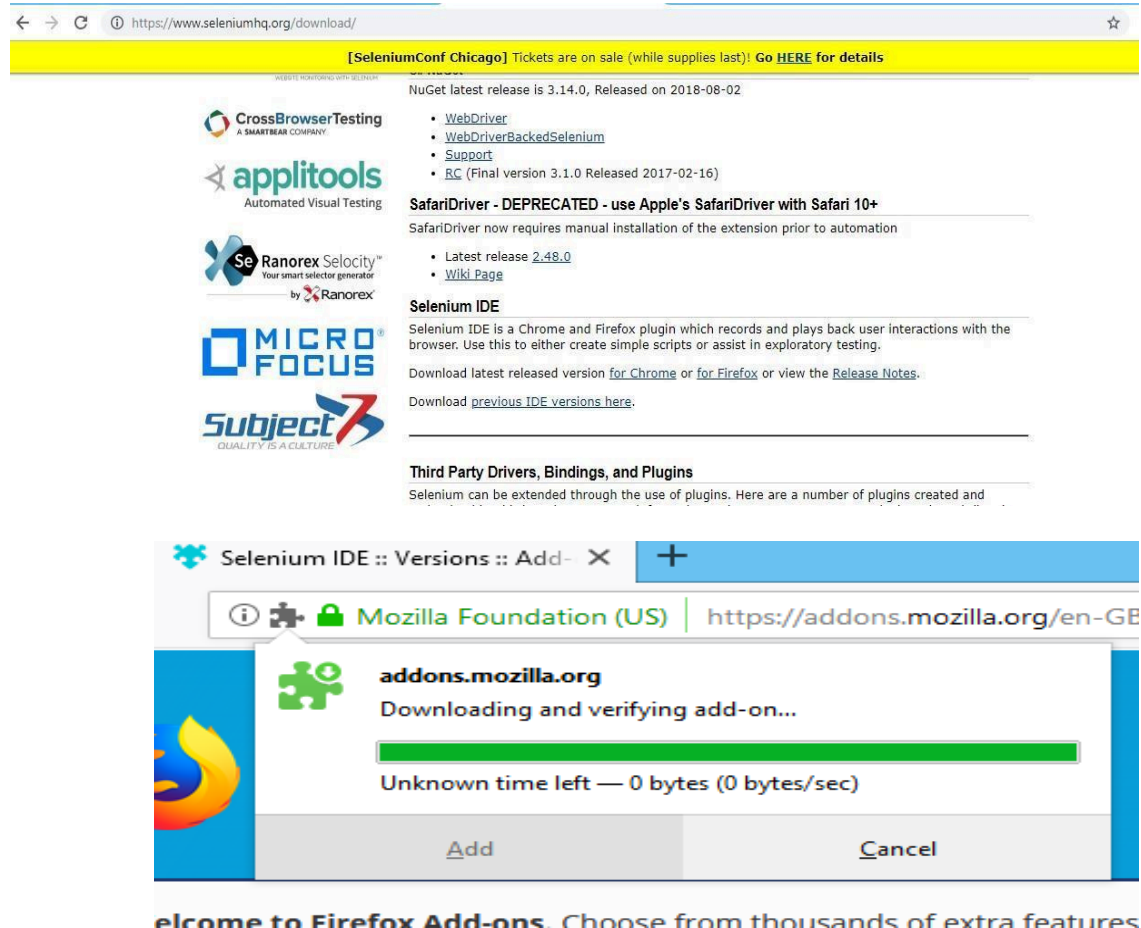
- Run a Test Suite- Click the Run All button to run all the test cases in the currently loaded test suite.
- Stop and Start -The Pause button can be used to stop the test case while it is running. The icon of this button then changes to indicate the Resume button. To continue click Resume.
- Stop in the Middle- You can set a breakpoint in the test case to cause it to stop on a particular command. This is useful for debugging your test case. To set a breakpoint, select a command, right-click, and from the context menu select Toggle Breakpoint.
- Start from the Middle- You can tell the IDE to begin running from a specific command in the middle of the test case. This also is used for debugging. To set a start point, select a command, right-click, and from the context menu select Set/Clear Start Point.
- Run Any Single- Command Double-click any single command to run it by itself. This is useful when writing a single command. It lets you immediately test a command you are constructing, when you are not sure if it is correct. You can double-click it to see if it runs correctly. This is also available from the context menu.
- Test Suite: A test suite is a collection of tests. Often one will run all the tests in a test suite as one continuous batch-job. When using Selenium- IDE, test suites also can be defined using a simple HTML file. The syntax again is simple. An HTML table defines a list of tests where each row defines the file system path to each test.

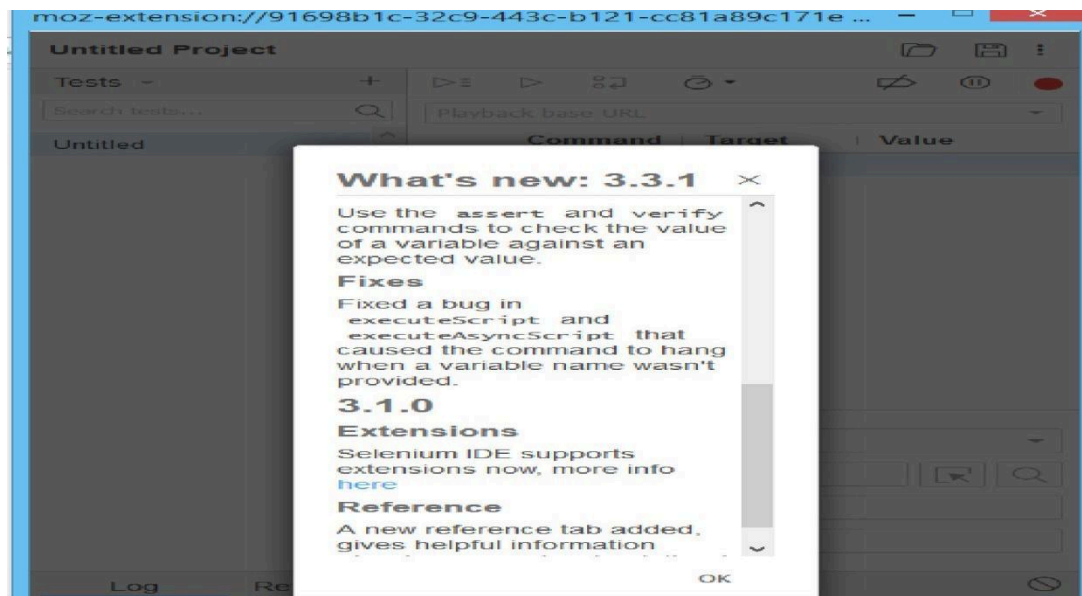
## **G. Installing the IDE**

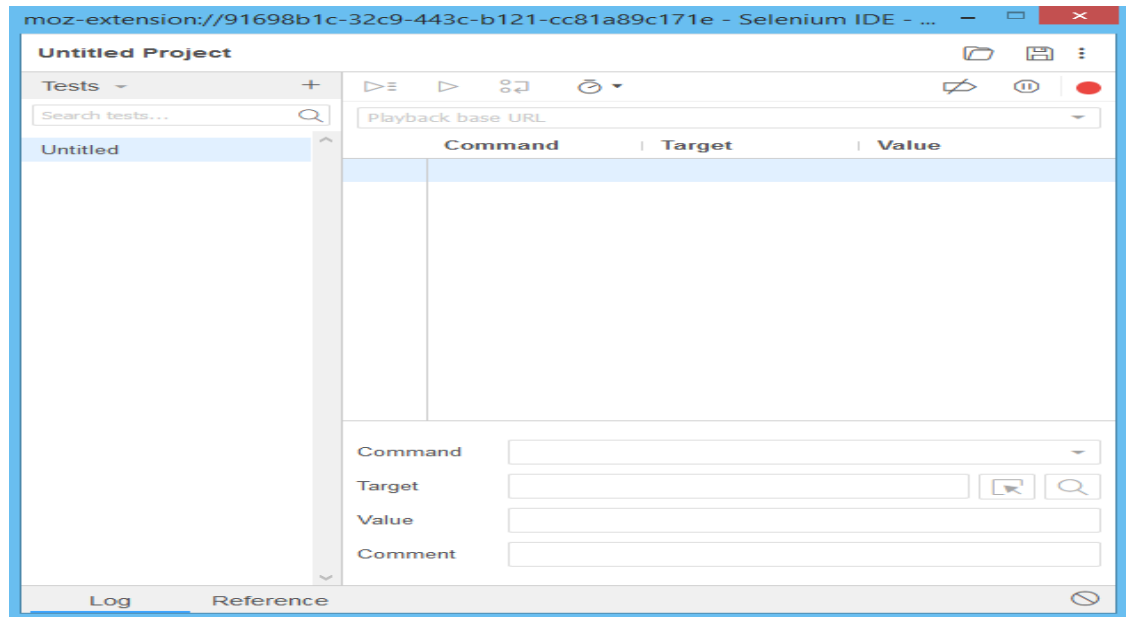
Using Firefox, first, download the IDE from the SeleniumHQ [downloads page](#) Firefox will protect you from installing addons from unfamiliar locations, so you will need to click „Allow“ to proceed with the installation, as shown in the following screenshot.

When downloading from Firefox, you“ll be presented with the following window.

- Select Install Now. The Firefox Add-ons window pops up, first showing a progress bar, and when the download is complete, displays the following.
- Restart Firefox. After Firefox reboots you will find the Selenium-IDE listed under the Firefox Tools menu.
- Opening the IDE: To run the Selenium-IDE, simply select it from the Firefox Tools menu. It opens as follows with an empty script-editing window and a menu for loading, or creating new test cases.
- Click on recording button and start performing testing. When finish with testing stop recording







Now, Write a test suite containing minimum 4 test cases.

#### **TC'S #1: Manual Steps:**

- Open (Example : Type [www.google.com](http://www.google.com))
- Type "Mumbai University" in the Google Search Input Box
- Click outside on an empty spot
- Click Search Button
- Verify the Text Present as "Mumbai University"
- Assert the Title as "Mumbai University - Google Search"
- Save the test case with .HTML Extension.

#### **TC'S #2: Manual Steps:**

- Open (Example : Type [www.google.com](http://www.google.com))
- Type "Selenium RC" in the Google Search Input Box
- Click outside on an empty spot
- Click Search Button
- Verify the Text Present as "Selenium RC"
- Assert the Title as "Selenium RC - Google Search"
- Save the test case with .HTML Extension.

### **TC'S #3: Manual Steps:**

- Open (Example : Type www.google.com)
- Type "Guru99" in the Google Search Input Box
- Click outside on an empty spot
- Click Search Button
- Verify the Text Present as "Guru99"
- Assert the Title as "Guru99"- Google Search"
- Save the test case with .HTML Extension.

### **TC'S #4: Manual Steps:**

- Open (Example : Type www.google.com)
- Type "Software testing Help" in the Google Search Input Box
- Click outside on an empty spot
- Click Search Button
- Verify the Text Present as "Software testing Help"
- Assert the Title as "Software testing Help "- Google Search"
- Save the test case with .HTML Extension.

### **Steps for creating test suite:**

1. Create more Tc's save each Test Case with <.html> extension.
2. Open Firefox
3. Open Tools → Selenium IDE
4. File → Open → new Test Suite
5. File → Open → Add Test cases
6. Add more test cases
7. Save Suite with <.Html> extensions. Run the test suite

## Practical : 2

### 2. Conduct a test suite for any two web

#### sites. Solution:

##### TC'S #1: Manual Steps:

- Open (Example : Type [www.google.com](http://www.google.com))
- Type "Mumbai University" in the Google Search Input Box
- Click outside on an empty spot
- Click Search Button
- Verify the Text Present as "Mumbai University"
- Assert the Title as "Mumbai University - Google Search"
- Save the test case with .HTML Extension.

##### TC#2: Manual Steps

- Open Firefox Web Browser
- In the address bar, Type <http://www.yahoo.com>
- In the search input button, Type " JVM's Mehta College"
- Click on the "Web Search" submit button
- Wait for Search Results to come on "http://search.yahoo.com"
- Verify "JVM's Mehta college" text is present anywhere in the search results:  
(Select and highlight anywhere in the search results page, " JVM's Mehta college " text is present.)
- Verify the browsers title has the value " JVM's Mehta college " - Search Results"
- Save the test case with .HTML Extension.

**Steps for creating test suite:**

- Create more Tc's save each Test Case with <.html> extension.
- Open Firefox
- Open Tools → Selenium IDE
- File → Open → new Test Suite
- File → Open → Add Test cases
- Add more test cases

Save Suite with <.Html> extensions and Run the test suite

## Practical : 3

### **3. Install Selenium server (Selenium RC) and demonstrate it using a script in Java/PHP**

#### **i.Introduction**

Selenium-RC is the solution for tests that need more than simple browser actions and linear execution. Selenium-RC uses the full power of programming languages to create more complex tests like reading and writing files, querying a database, emailing test results. You'll want to use Selenium- RC whenever your test requires logic not supported by Selenium-IDE. What logic could this be? For example, Selenium-IDE does not directly support:

- Condition statements
- Iteration
- logging and reporting of test results
- Error handling, particularly unexpected errors
- Database testing
- Test case grouping
- Re-execution of failed tests
- Test case dependency
- Screenshot capture of test failures

Although these tasks are not supported by Selenium directly, all of them can be achieved by using programming techniques with a language-specific Selenium- RC client library

#### **ii.Installation of Eclipse**

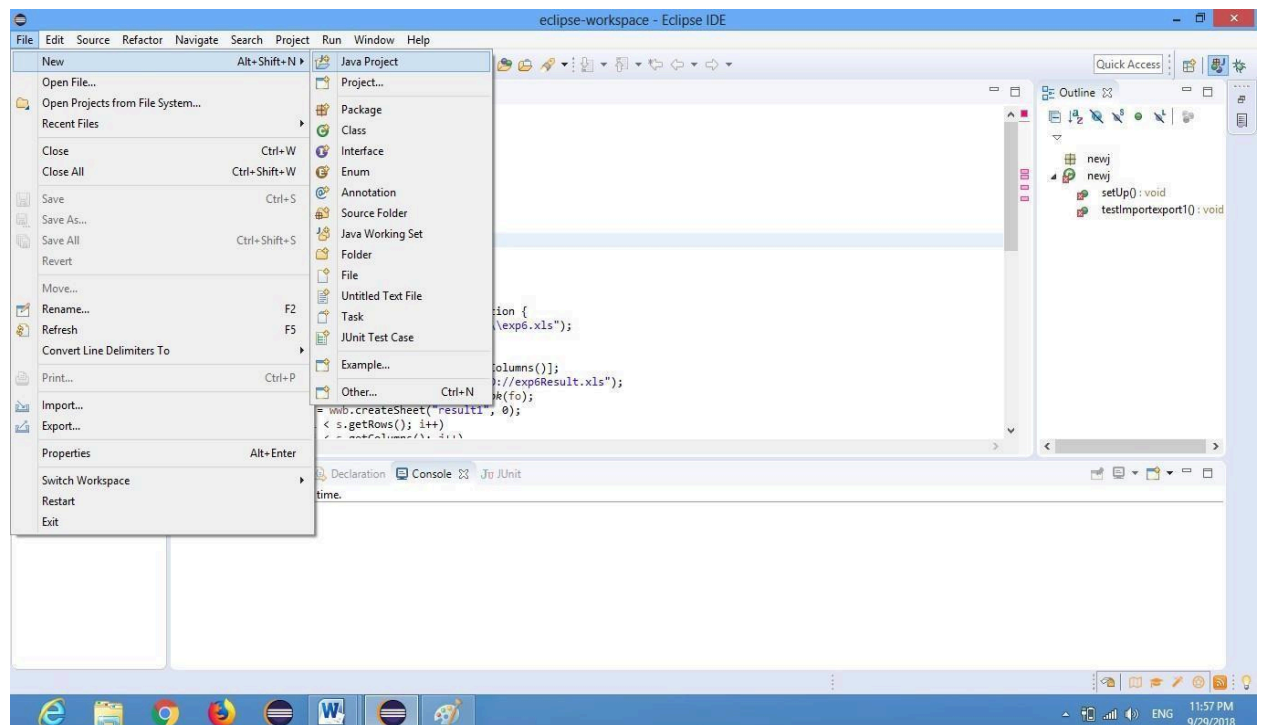
Eclipse (@ [www.eclipse.org](http://www.eclipse.org)) is an *open-source* Integrated Development Environment (IDE) supported by IBM. Eclipse is popular for Java application development (Java SE and Java EE) and Android apps. It also supports C/C++, PHP, Python, Perl, and other web project developments via extensible plug-ins. Eclipse is cross-platform and runs under Windows, Linux and Mac OS.

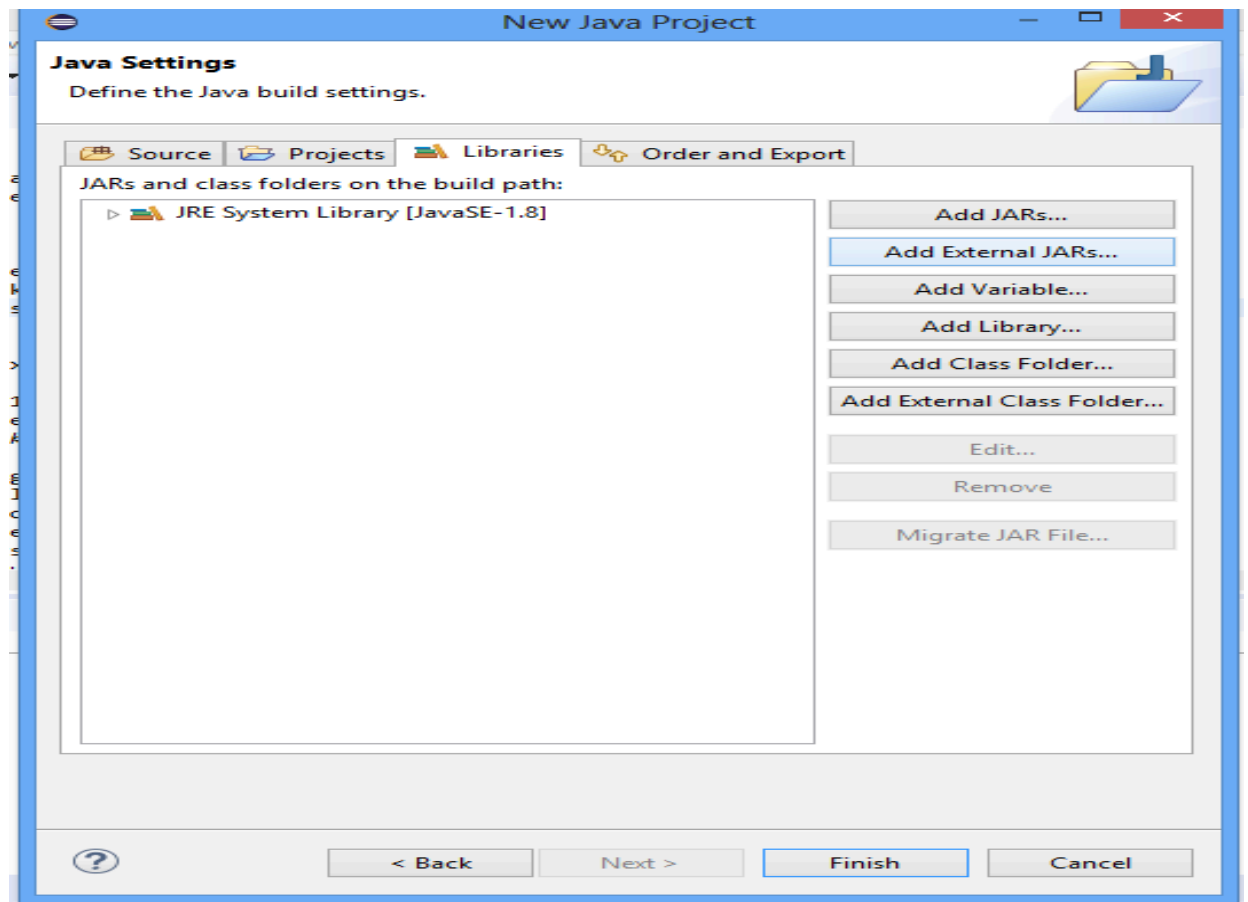
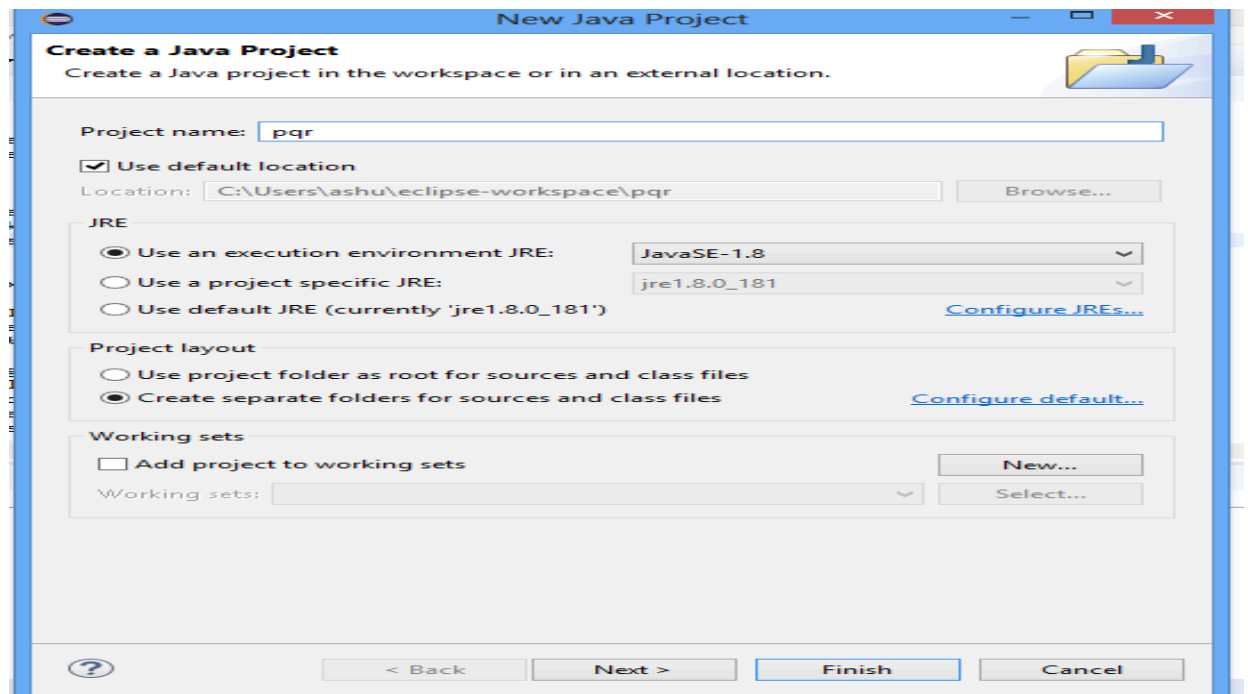


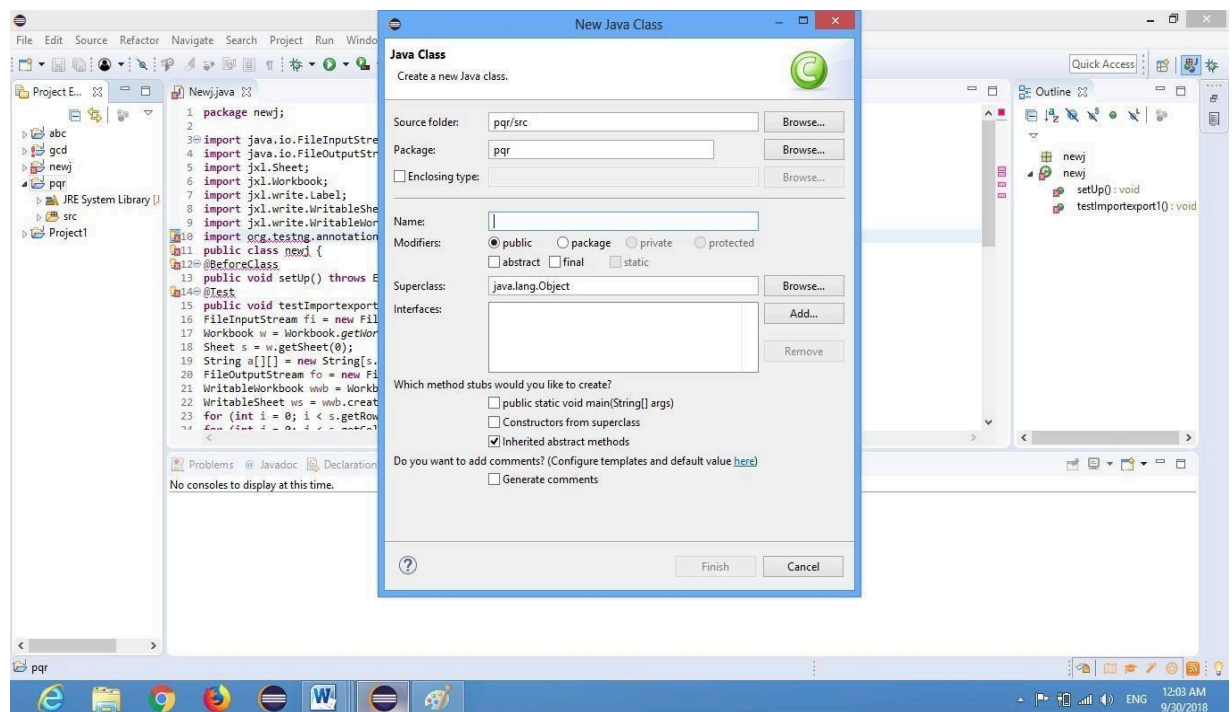
Install Eclipse for Java Developers using below steps:

- Go to URL – <http://www.eclipse.org/downloads/>
- Select Eclipse IDE for Java Developers (Click on Windows 32 bit platform)
- Click on OK button and save to a local drive (i.e. C: or D:, etc)
- Unzip the downloaded zip file and rename that to **Eclipse**
- Create one more folder “Eclipse-Workspace” (i.e. C:Eclipse- Workspace)in the same drive where Eclipse is unzipped and renamed.
- Create Eclipse desktop shortcut (go to C:Eclipse folder → right click Eclipse.exe and then click on “desktop create shortcut”)
- Create project in Eclipse using below steps:
- Click on File->New->Java project. Give name java project and click on next
- Go to libraries Add external jar files if required. Click on finish.
- Right Click on created project Go to src and add java class. Click on

finish. To execute this steps screenshots are also provide below:







### iii. Selenium Server:

To download and run selenium server we need to follow steps given below:

1.Download Selenium Standalone Server from this link

<https://www.seleniumhq.org/download/>

The screenshot shows the SeleniumHQ website. The header includes the SeleniumHQ logo, a search bar, and navigation links for Projects, Download, Documentation, Support, and About. The left sidebar has links for Selenium Downloads, Previous Releases, Source Code, and Maven Information. The main content area is titled 'Downloads' and contains several sections: 'Selenium Standalone Server' with a description and a download link for version 3.14.0; 'The Internet Explorer Driver Server' with a description and a download link for version 3.14.0; and 'Selenium Client & WebDriver Language Bindings' with a description. There are also links for 'Donate to Selenium' and 'BrowserStack'.

2. Download Selenium Client server from this link

<http://www.java2s.com/Code/Jar/s/Download selenium java client driver 101jar.html>

The screenshot shows the Java2S.com website. The main content area is titled 'Download' and contains a section for 'Jar File Download'. The download link is for 'selenium-java-client-driver-1.0.1.jar' (50 k). The download link is highlighted in blue. Below the link, there is a list of class files or Java source files contained in the jar file.

3. To start selenium server open cmd and hit command given below  
 java -jar  
 C:\Users\Admin\Desktop\selenium-server-standalone-3.14.0.jar  
 -port 4444

```
Command Prompt - java -jar C:\Users\ashu\Desktop\testing\exp4\selenium-ser...
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\ashu>java -jar C:\Users\ashu\Desktop\testing\exp4\selenium-server-standalone-3.14.0.jar -port 4444
00:25:57.051 INFO [GridLauncherU3.launch] - Selenium build info: version: '3.14.0', revision: 'aaccce0'
00:25:57.051 INFO [GridLauncherU3$1.launch] - Launching a standalone Selenium Server on port 4444
2018-09-30 00:25:57.208:INFO::main: Logging initialized @544ms to org.seleniumhq.jetty9.util.log.StdErrLog
00:25:57.755 INFO [SeleniumServer.boot] - Selenium Server is up and running on port 4444
```

## Code for gcd.html

```
<html>
<head>
<title> GCD </title>
<script type="text/javascript">
function gcd()
{
var x,y;
x=parseInt(document.myform.n1.value);
y=parseInt(document.myform.n2.value);
while(x!=y)
{ if(x>y)
x=x-y;
else
y=y-x;
}
document.myform.result.value=x;
}
</script>
</head>
<body>
<h1 align="center"> Program to calculate gcd of two numbers </h1>
<hr color="red">
<center>
Enter two numbers :
<form name="myform">
Number 1 : <input type="text" name="n1" value=""> <br> <br>
Number 2 : <input type="text" name="n2" value=""> <br> <br>
```

```
<input type="button" name="btn" value="Get GCD" onClick="gcd()"> <br> <br>
GCD is : <input type="text" name="result" value="">
</form>
</body>
</html>
```

### **Source Code:**

```
package pracgcd;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class GCD {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.setProperty("webdriver.gecko.driver","C://Users//Admin//Desktop//geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.get("file:///C://Users//Admin//Desktop//gcd.html" );
        driver.manage().window().maximize();
        driver.findElement(By.id("n1")).sendKeys("2");
        driver.findElement(By.id("n2")).sendKeys("14");
        //driver.findElement(By.id("txt1")).sendKeys("49");
        driver.findElement(By.name("gcd")).click();
        String result=
        driver.findElement(By.name("result")).getAttribute("name=result");
        System.out.println("the gcd is:"+ result);
    }
}
```



## Program to calculate gcd of two numbers

---

Enter two numbers :

Number 1 :

Number 2 :

GCD is :



Practical : 4

**4. Write and test a program to login a specific web page Solution:**

**Source Code:**

```
package prackage;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Pack {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.setProperty("webdriver.gecko.driver","C://Users//Admin//Desktop//geckodriver.exe");
        WebDriver driver=new FirefoxDriver();
        driver.get("https://mail.google.com");


        driver.findElement(By.id("identifierId")).sendKeys("jvmcollegeairoli@gmail.com");
        // driver.findElement(By.id("Passwd")).sendKeys("yourPassword");
        driver.findElement(By.id("signIn")).click();

    }

}
```







## Sign in

to continue to Gmail

Email or phone

[Forgot email?](#)

Not your computer? Use a Private Window to sign in.  
[Learn more](#)

[Create account](#)[Next](#)

English (United Kingdom)  [Help](#) [Privacy](#) [Terms](#)



Practical : 5

**5. Write and test a program to update 10 student records into table into Excel file.**

**Solution:**

### **Introduction to TestNG:**

TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionality that make it more powerful and easier to use, such as:

- Annotations.
- Run your tests in arbitrarily big thread pools with various policies available (all methods in their own thread, one thread per test class, etc...).
- Test that your code is multithread safe.
- Flexible test configuration.
- Support for data-driven testing (with `@DataProvider`).
- Support for parameters.
- Powerful execution model (no more Test Suite).
- Supported by a variety of tools and plug-ins (Eclipse, IDEA, Maven, etc...).
- Embeds bean Shell for further flexibility.
- Default JDK functions for runtime and logging (no dependencies).
- Dependent methods for application server testing.

TestNG is designed to cover all categories of tests: unit, functional, end-to-end, integration, etc...

- Installing TestNG in eclipse
- Select Help / Software updates / Find and Install.
- Search for new features to install.

- New remote site.

For Eclipse 3.4 and above, enter <http://beust.com/eclipse>.

For Eclipse 3.3 and below, enter

<http://beust.com/eclipse1>. Make sure the check box

next to URL is checked and click Next. Eclipse will

then guide you through the process.

Launching your tests in Eclipse

We finished writing our tests, now how can we run them?

You can launch TestNG from the command line, using a Eclipse plug-in or even programmatically. We are going to use the Eclipse plugin. Follow the steps described on the official TestNG documentation [over here](#)

If you installed TestNG correctly, you will see this menu when you right click on the XML file:

Click on “Run as TestNG Suite” and your test will start running. You will then see this nice results tree:

## **Selenium Tests with Microsoft Excel**

Parameterizing a test from external sources such as Microsoft Excel is always Recommended in order to handle large amount of test data. To read data from Excel, we need APIs which support opening file, reading data, and writing data into Excel. We should know various classes and methods which support above mentioned operations. In this post, let us try to figure out which is the API that supports all the activities we need to do during execution of a test. Jxl.jar is an open source Java API which supports read Excel spreadsheets and to write into Excel spreadsheets. Below are some of the operations that we can handle with this API.

- Read data from Excel spreadsheet
- Read and write formulas into spreadsheets
- Generate spreadsheets
- Supports formatting of font, number, and date
- Supports coloring of cells

To access the methods and classes provided by this API inside Eclipse we need to add this JAR file to the Java Build Path. (I have explained steps to add external Jar files to Java Build Path in previous posts)

Download the jxl.jar from “<http://jexcelapi.sourceforge.net/>” Add the JAR file to Java Build Path

### Source Code:

```
package excelproj;
import java.io.File;
import java.io.IOException;
import java.util.Locale;
import jxl.CellView;
import jxl.JXLException;
import jxl.Workbook;
import jxl.WorkbookSettings;
import jxl.format.UnderlineStyle;
import jxl.write.Formula;
import jxl.write.Label;
import jxl.write.Number;
import jxl.write.WritableCellFormat;
import jxl.write.WritableFont;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import jxl.write.WriteException;
import jxl.write.biff.RowsExceededException;

public class Writeexcel {

    private static final String Firstcolumn = null;
    //public static void main(String[] args) {
        // TODO Auto-generated method stub
```

```

        private WritableCellFormat timesBoldUnderline;
        private WritableCellFormat times;
        private String inputFile;
        public void setOutputFile(String inputFile) {
            this.inputFile = inputFile;
        }
        public void write() throws IOException, WriteException { File file = new
        File(inputFile);
        WorkbookSettings wbSettings = new WorkbookSettings();
        wbSettings.setLocale(new Locale("en", "EN"));
        WritableWorkbook workbook = Workbook.createWorkbook(file, wbSettings);
        workbook.createSheet("Report", 0);
        WritableSheet excelSheet = workbook.getSheet(0); createLabel(excelSheet);
        createContent(excelSheet);
        workbook.write(); workbook.close();
        }
        private void createLabel(WritableSheet sheet)
        throws WriteException {
            // Lets create a times font
            WritableFont times10pt = new WritableFont(WritableFont.TIMES, 10);
            // Define the cell format
            times = new WritableCellFormat(times10pt);
            // Lets automatically wrap the cells
            times.setWrap(true);
            // Create a bold font with underlines
            WritableFont times10ptBoldUnderline = new WritableFont(
            WritableFont.TIMES, 10, WritableFont.BOLD, false,
            UnderlineStyle.SINGLE);
            timesBoldUnderline = new WritableCellFormat(times10ptBoldUnderline);
            // Lets automatically wrap the cells
            timesBoldUnderline.setWrap(true);
            CellView cv = new CellView(); cv.setFormat(times);
            cv.setFormat(timesBoldUnderline); cv.setAutosize(true);
            // Write a few headers
            addCaption(sheet, 0, 0, "Student Name");
            addCaption(sheet, 1, 0, "Subject 1");
            addCaption(sheet, 2, 0, "subject 2");
            addCaption(sheet, 3, 0, "subject 3");
        }
        private void createContent(WritableSheet sheet) throws WriteException,

```

```

RowsExceededException {
    // Write a few number
    for (int i = 1; i < 10; i++) {
        //First column
        addLabel(sheet, 0, i, "student " + i);
        //Second column
        addNumber(sheet, 1, i, ((i*i)+10));
    }
    addNumber(sheet, 1, i, ((i*i)+2));
    addNumber(sheet, 2, i, ((i*i)+3));
    addNumber(sheet, 3, i, ((i*i)+4));
}

private void addCaption(WritableSheet sheet, int column, int row, String s)
throws RowsExceededException, WriteException { Label label;
label = new Label(column, row, s, timesBoldUnderline); sheet.addCell(label);
}

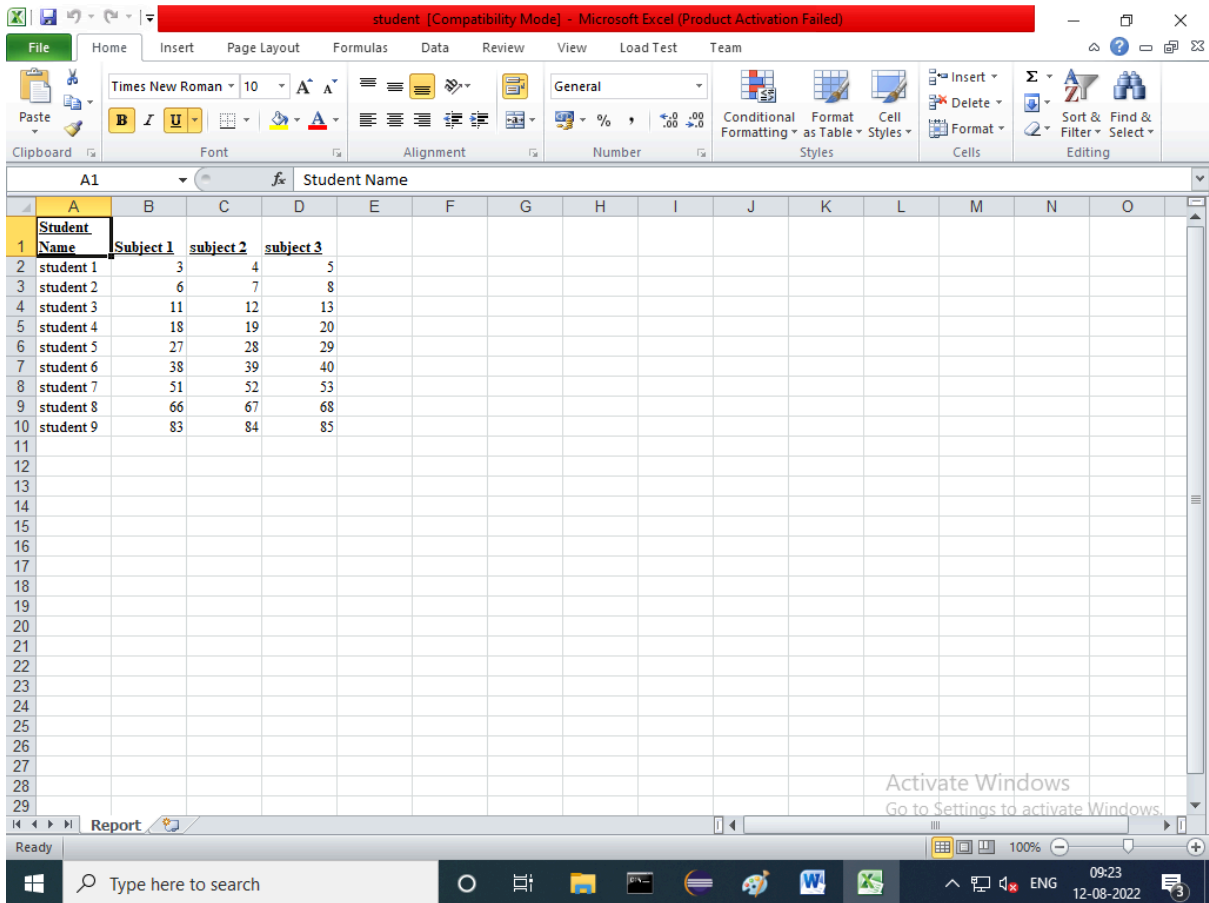
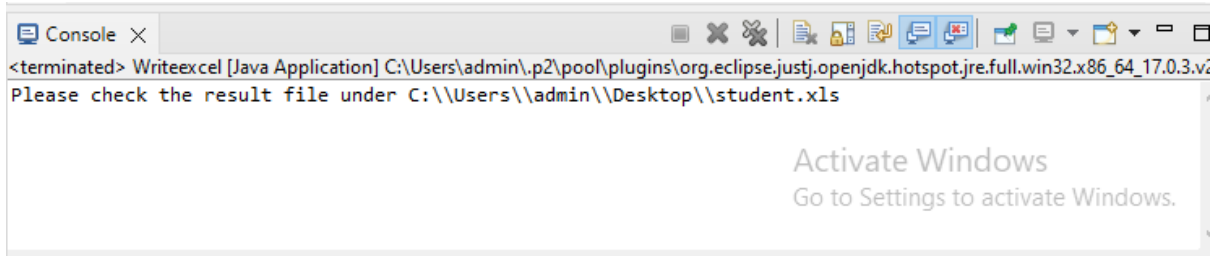
private void addNumber(WritableSheet sheet, int column, int row,
Integer integer) throws WriteException, RowsExceededException { Number
number;
number = new Number(column, row, integer, times); sheet.addCell(number);
}

private void addLabel(WritableSheet sheet, int column, int row, String s)
throws WriteException, RowsExceededException { Label label;
label = new Label(column, row, s, times); sheet.addCell(label);
}

public static void main(String[] args) throws JXLException, IOException {
    //TODO Auto-generated void method stub
    Writeexcel test = new Writeexcel();
    test.setOutputFile("C:\\Users\\admin\\Desktop\\student.xls");

    test.write();
    System.out.println("Please check the result file under
C:\\Users\\admin\\Desktop\\student.xls ");
}
}

```



Practical : 6

**6. Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).**

**Solution:**

```
package Excelwriter;
import java.io.File;
import java.io.IOException;
import jxl.Cell;
import jxl.CellType;
import jxl.Sheet;
import jxl.Workbook;
import jxl.read.biff.BiffException;

public class excel_read_data {
    private String inputFile;
    public void setInputFile(String inputFile) {
        this.inputFile = inputFile;
    }
    public void read() throws IOException {
        File inputWorkbook = new File(inputFile); Workbook w;
        boolean flag=false;
        int count=0;
        try {
            w = Workbook.getWorkbook(inputWorkbook);
            // Get the first sheet
            Sheet sheet = w.getSheet(0);
            // Loop over first 10 column and lines
```

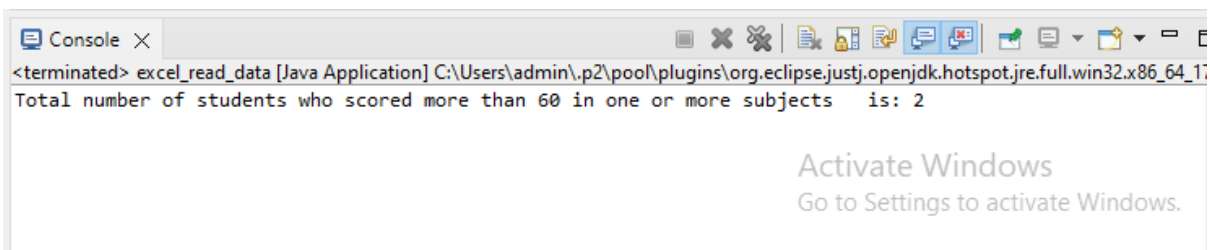


```

        for (int j = 0; j < sheet.getRows(); j++) {
        for (int i = 0; i < sheet.getColumns(); i++) { Cell cell = sheet.getCell(i,
j);
        if (cell.getType() == CellType.NUMBER) {
        if(Integer.parseInt(cell.getContents())>60){ flag = true;
        if(flag == true){
        count++; flag=false;
        }
        break;
        }
        }
        }
        }
        System.out.println("Total number of students who scored more than 60 in
one or more subjects  is: " +count);
        }
        catch (BiffException e) { e.printStackTrace();
        }
        }
        public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        excel_read_data test = new excel_read_data();
        test.setInputFile("C:\\Users\\admin\\Desktop\\student.xls");
        test.read();

        }
    }

```



Practical : 7

**7. Write and test a program to provide total number of objects present / available on the page.**

**Solution:**

**Source Code:**

```
package prac7;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class prac_7 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.setProperty("webdriver.gecko.driver", "C://Users//Admin//Desktop//geck
        odriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.get("http://google.co.in/");
```

```

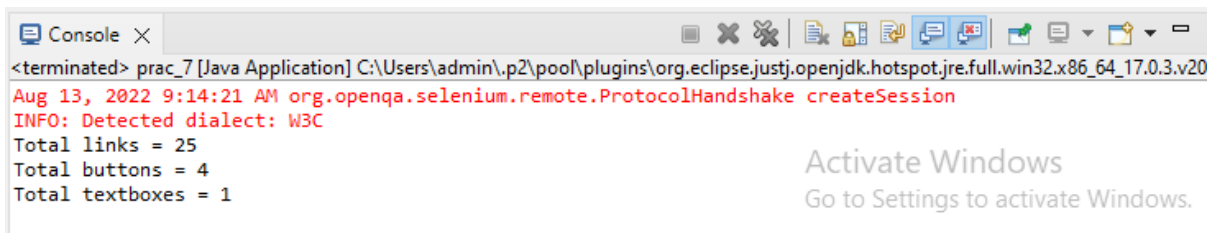
        java.util.List<WebElement> links =
driver.findElement(By.tagName("a")); System.out.println("Total links = "+
links.size());
//          java.util.List<WebElement> buttons =
driver.findElement(By.tagName("button"));
        List<WebElement> buttons =
driver.findElement(By.xpath("//input[@type='button' or @type='submit']"));
        System.out.println("Total buttons = "+ buttons.size());
List<WebElement> textboxes =
        driver.findElement(By.xpath("//input[@type='text' or
@type='password']"));

//          java.util.List<WebElement> textboxes =
driver.findElement(By.xpath("//input[@type='text'[@class='inputtext']]"));
        System.out.println("Total textboxes = " + textboxes.size());

    }

}

```



The screenshot shows a console window titled "Console" with the following output:

```

<terminated> prac_7 [Java Application] C:\Users\admin\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20
Aug 13, 2022 9:14:21 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Total links = 25
Total buttons = 4
Total textboxes = 1

```

On the right side of the console window, there is a watermark that says "Activate Windows" and "Go to Settings to activate Windows."

Practical : 8

**8. Write and test a program to get the number of items in a list / combo box.**

**Solution:**

- Write this java test code in eclipse. before running this code makes sure your selenium RC server is running)
- Write Combocount.html as shown below and save under desktop

```
<html>  
<body>  
<select>  
<option>Volvo</option>
```

```
<option>Express</option>
<option>Mercedes</option>
<option>RajaHamsa</option>
</select>
</body>
</html>
```

- Write the below code in eclipse
- Make sure that the path of your combocount.html file must be correct for your system , in this example it is

file:///C:/Users/Desktop/combocount.html

```
package p8;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter; import java.io.IOException;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
public class testCombobox {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.setProperty("webdriver.gecko.driver","C://Users//Admin//
            Desktop//geckodriver.exe");

        WebDriver driver = new FirefoxDriver();

        driver.get("file:///C://Users//admin//Desktop//combocount.html");

        Select se = new Select(driver.findElement(By.xpath("//select")));

        List<WebElement> l = se.getOptions();
        l.size();
        System.out.println("Number of Items: " + l.size());
```

```
}  
  
}
```

